Яблонцева А.Д.
Хакасский государственный университет им. Н.Ф. Катанова, Россия, г. Абакан

ОБЗОР ТЕХНОЛОГИИ TESSERACT 4.0 И ТИПИЧНЫЕ ПРОБЛЕМЫ РАСПОЗНАВАНИЯ ТЕКСТОВ

Аннотация

В работе описывается движок OCR с открытым исходным кодом. Разбираются особенности Tesseract 4.0. Описываются проблемные ситуации распознавания текстов и приводятся рекомендации для их устранения.

Ключевые слова: технология оптического распознавания символов
Keywords: optical character recognition technology, LSTM, Tesseract

Научный руководитель Голубничий А.А.
старший преподаватель кафедры ПОВТиАС,
ХГУ им. Н.Ф. Катанова, Россия, г. Абакан

Tesseract – является программой, что была разработана фирмой Hewlett-Packard [1] в период с середины 1980 по 1990 года, в августе 2006 года её купила компания Google. Программа представляет движок OCR (технология оптического распознавания символов) с открытым исходным кодом. Его можно использовать через командную строку или с помощью API для извлечения печатного текста из изображений [2]. В Tesseract 4.0 – добавлен движок OCR [3] на основе нейронных сетей LSTM. Это увеличило точность распознавания текста на изображениях документов, но учитывая, что нейронные сети требуют больше обучающих данных, они работают медленнее чем базовый Tesseract. Для языков, основанных на латинице, имеющиеся данные модели были обучены примерно на 400000 текстовых строках, охватывающих примерно 4500 шрифтов. По обучению Tesseract 4.00 занимает от нескольких дней до пары недель.

К техническим требованиям для запуска Tesseract 4.0.0 необходиим многоядерный (4 ядра будет достаточно) компьютер с поддержкой OpenMP и Intel Intrinsics для расширений SSE/AVX. Он так же по-прежнему будет работать на любом устройстве с достаточным объемом памяти, но чем выше уровень процессора, тем быстрее он будет работать. Графический процессор не требуется. Использование памяти можно контролировать с помощью параметра командной строки – max_image_MB, но вероятно потребуется как минимум 1 ГБ памяти сверх того, что занимает ОС.

С Tesseract легче работать используя тонкую настройку. Это означает, что имея уже обученный язык, проще тренироваться на конкретных дополнительных данных. Можно так же обучить новому языку, для этого нужно отрезать верхний слой или какое-то произвольное количество слоёв от сети и переобучить, используя новые данные [4].

LSTM отлично подходят для обучения последовательностям, но сильно замедляются, когда количество состояний слишком велико. Существуют эмпирические результаты, которые показывают, что лучше попросить LSTM изучить длинную последовательность, чем короткую последовательность из многих классов, поэтому для сложных скриптов (скрипты хань, хангыль и индийский язык) лучше перекодировать каждый символ как короткую последовательность кодов из небольшого числа классов, чем из большого набора классов.

Как и в случае с базовым Tesseract, есть выбор между рендерингом синтетических обучающих данных из шрифтов или маркировкой некоторых ранее существовавших изображений (например, древних рукописей).В любом случае, требуемым форматом по-прежнему является пара файлов tiff / box, за исключением того, что поля должны закрывать только текстовую строку, а не отдельные символы [5].

При запуске обучения могут возникать различные сообщения об ошибках, некоторые из которых могут быть важными, а другие – нет. Текстовая строка для обучающего изображения не может быть закодирована с использованием данного единого набора символов, по следующим причинам:

- В тексте есть не представленный символ, к примеру знак британского фунта, которого нет в кодировке.
- Случайный непечатаемый символ (например, табуляция или управляющий символ) в тексте.
- В тексте присутствует не представленная индийская графема/акшара.

В любом случае это приведет к тому, что система проигнорирует обучающий образ.

Tesseract выполняет различные операции обработки изображений внутри (используя библиотеку Leptonica) перед тем, как выполнять собственное распознавание текста. Как правило, он очень хорошо справляется с этим, но неизбежны случаи, когда он будет совершать ошибки, что может привести к значительному снижению точности.

Tesseract обрабатывает изображение, используя переменную конфигурации tessedit_write_images to true (или используя configfile get.images) при запуске. Если полученный tessinput.tif файл выглядит проблематичным, тогда лучше попробовать некоторые из операций обработки изображения, прежде чем передавать изображение в Tesseract [6].

Жирные или тонкие символы могут повлиять на распознавание деталей и снизить точность распознавания. Многие программы обработки изображений позволяют расширять и размывать края символов на общем фоне. Сильное вытекание чернил из исторических документов можно компенсировать с помощью техники эрозии. Эрозию [7] можно использовать для уменьшения символов до их нормальной структуры глифов. Например, фильтр «Распространение значения» в GIMP может создать размытие дополнительных жирных исторических шрифтов за счет уменьшения нижнего порогового значения. Пример представлен на рисунке 1.

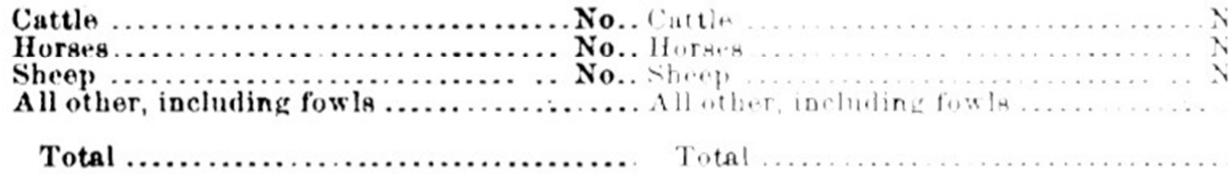

Рисунок 1 – Пример применения эрозии.

Качество линейной сегментации Tesseract значительно снижается [8], если страница слишком перекошена, что серьезно влияет на качество распознавания текста. Чтобы решить эту проблему, достаточно повернуть изображение страницы так, чтобы текстовые строки были горизонтальными (рисунок 2).

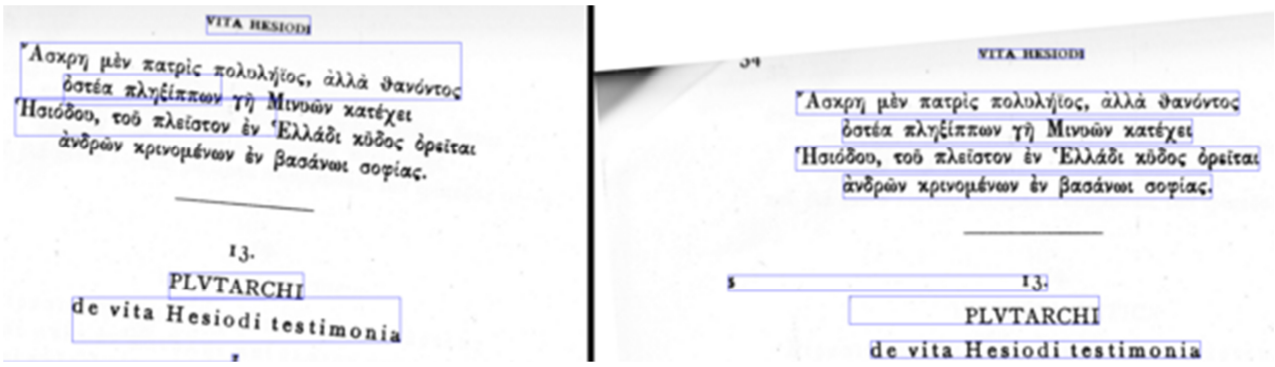

Рисунок 2—Повышение качества линейной сегментации

Заключение

Построение системы эффективного распознавания символов представляет собой достаточно сложную задачу. Разработанная более 30 лет назад система распознавания символов Tesseract постоянно развивается и дополняется новыми функциями, существующая версия системы Tesseract 4.0, в значительной степени решает поставленные перед ней задачи, однако, развитие новых технологий и подходов в данной области не заставляет себя ждать.

Литература

1. О компании |HP Россия [электронный ресурс] URL https://clck.ru/Wu86K (дата обращения 13.08.2021)
2. How to use the tools provided to train Tesseract 4.00 | tessdoc [электронный ресурс] URL https://clck.ru/Wu86P (дата обращения 13.08.2021)
3. Использование оптического распознавания символов [электронный ресурс] URL https://clck.ru/Wu86Z (дата обращения 13.08.2021)
4. Tesseract User Manual | tessdoc [электронный ресурс] URL https://clck.ru/Wu86o (дата обращения 13.08.2021)
5. Improving the quality of the output | tessdoc [электронный ресурс] URL https://clck.ru/Wu86u (дата обращения 13.08.2021)
6. Error during LSTM Training | tessdoc [электронный ресурс] URL https://clck.ru/Wu86u (дата обращения 13.08.2021)
7. Назначение морфологической обработки бинарных изображений [электронный ресурс] URL https://clck.ru/Wu86y (дата обращения 13.08.2021)
8. Сегментация изображения - Image segmentationВикипедия site:wikichi.ru [электронный ресурс] URL https://clck.ru/Wu87G (дата обращения 13.08.2021)